Terceiro Trabalho Prático de Sistemas de Operação

Simulador de escalonador

Ano Lectivo 2007/2008

1 Introdução

Nesta aula pretende-se ilustrar a matéria da aula teórica sobre algoritmos de escalonamento do CPU. Vamos utilizar um simulador do escalonamento do CPU, escrito em Java, que tem como base um da autoria de Ray Ontko (este último faz parte do material de apoio ao livro "Modern Operating Systems " de A.S. Tanenbaum).

Esta versão do simulador inclui dois algoritmos de escalonamento que simulam a execução que um lote de processos com várias características no que diz respeito ao tempo de CPU necessário, operações de I/O e sua duração. O resultado da simulação é um relatório onde aparece um sumário dos tempos de execução para cada processo e um traço das execuções e bloqueios que ocorrerem ao longo do tempo.

2 Simulador

Obtenha o código do simulador e respectiva documentação. A documentação inclui um manual de instalação e outro de utilização. Note que estes dizem respeito ao simulador original, o que pode em alguns casos não corresponder exactamente ao simulador fornecido, em particular na descrição dos processos.

Compile. Pode executar dando comandos na forma:

`java Scheduling` *fich_configuração*

O *fich_configuração* descreve os processos. Experimente usar alguns dos ficheiros de extensão .conf incluidos no arquivo.

Ficheiro de configuração

O ficheiro de configuração descreve a simulação a efectuar, em particular, os vários processos que serão executados. Este vem descrito no manual de utilização, mas a versão fornecida tem algumas funcionalidades extra. Resumidamente, neste ficheiros indica-se o número de processos a executar, o tempo de CPU que estes necessitam para completar, o intervalo de tempo entre operações de I/O e a duração dessas operações. A simulação terminará quando todos os processos forem completamente executados ou então quando um tempo limite for atingido.

Resultados

Após cada simulação são produzidos os dois ficheiros seguintes:

`Summary-Processes`: traço da execução onde aparece uma linha por cada alteração de estado dos processos. Estas podem indicar que um novo processo começou ou retomou a execução, que o processo corrente se bloqueou ou terminou. Quando não existe nenhum processo para executar, o CPU fica desocupado (idle) e tal é representado por um processo fictício de identificador -1. O tempo apresentado à esquerda permite ter uma ideia da evolução do sistema e de cada processo, em particular o tempo real que cada processo necessitou para concluir a sua execução.

`Summary-Results`: este descreve brevemente os processos que foram executados e quantas vezes cada um se bloqueou. Em particular aparece o processo de identificador -1 para indicar o tempo em que o CPU ficou desocupado.

Algoritmos fornecidos

`SchedulingAlgorithm.java`: implementa um escalonador de um sistema com multiprogramação e preempção, seguindo-se uma ordem Round-Robin. O time-slice é indicado pelo valor da variável `quantum`.

`SchedulingAlgorithm1.java`: implementa um escalonador para um sistema sem multiprogramação, puramente sequencial.

Para testar cada um destes algoritmos terá de alterar no `main()` em `Scheduling.java` a chamada do algoritmo de simulação. Poderá ser:

```
result = SchedulingAlgorithm.Run(runtime, processVector, result);
```

ou

```
result = SchedulingAlgorithm1.Run(runtime, processVector, result);
```

3 Trabalho a realizar

Pretende-se comparar vários algoritmos de escalonamento usando este simulador. Dois serão os fornecidos e outros dois os que irá implementar de acordo com a secção seguinte.

Alterações a efectuar

Implemente as seguintes alterações aos algoritmos fornecidos:

- baseado no algoritmo em SchedulingAlgorithm, altere para que um processo ao terminar uma operação de I/O (ou seja quando se desbloqueia) fique à frente na fila *ready* em vez de ficar no fim.
- baseado no algoritmo em SchedulingAlgorithm1, altere este para contemplar multiprocessamento. Tal significa que, quando um processo se bloqueia o próximo da fila *ready* deve continuar a execução. Note que se pretende continuar a atender os processos segundo a ordem de chegada (FCFS).

Resultados a apresentar

Deve, para cada algoritmo executar o simulador para todas as configurações definidas e obter os valores do tempo real que cada processo demora até ser concluido e o tempo até todos os processos serem executados. Estes valores devem permitir calcular para cada algoritmo a taxa de trabalho concluido (*throughput*) e o tempo médio de resposta (*turnaround*). Verifique também o tempo em que o CPU fica desocupado para cada caso.